

MORE ON THE SURVEY

by Lynn Rudow

This month we'll give you a more detailed account of the survey results. After you have read these, if you have any comments, please take time to write to us.

We received 39 responses to the survey. Out of 149 current members, that's a 26.2% response rate. For your reference, the MUG's highest membership was somewhere between 350 and 400, some three or four years ago. We (Lynn & I) are doing something wrong. There certainly are more Micropolis and Vector Graphic computers now than there were four years ago.

1. What type of computer(s) do you use?

All respondents did not answer all question, but they all answered this one. For the 39 responses, we determined that there were 78 total computers, or exactly 2.0 computers per member. The following percentages are based on the number of total computers, i.e., 2.5% of the 78 computers were Altairs. Since the number of owners is 50% of the number of computers, you can double the percentage given to get the percent of owners having the specified system. That is, 5.0% (2.5*2) of the members have an Altair system.

Altair	02.5%
Apple	03.8%
Commadore	01.3%
Calif. Comp. Sys.	01.3%
CDS	01.3%
Compal	01.3%
Compupro	02.5%
Cromenco	01.3%
Exidy	01.3%
IBM-or compatible	19.2%
Imsai	02.5%
Kaypro 8-bit	01.3%
P-E 3205A	01.3%
S-100	05.1%
SOL	02.5%
Tektronix	01.3%
Televideo	01.3%
Timex Sinclair 1000	01.3%
TRS-80	01.3%
TRS-100	02.5%
VG-B	16.6%
VG-MZ	03.8%
VG-3	03.8%
VG-4	06.4%
VG-3xxx	02.5%
VG-4xxx	01.3%
VG-5xxx	05.1%
VG-VSX	01.3%
Xitan	01.3%

(continued on page 5)

CP/M 3.0 ON MICROPOLIS MOD II

by Melvin M. Dalton

7826 W 80th St., Playa del Rey, CA 90293

DO YOU WANT IT?

Why would you want 3.0 instead of your good old stand-by 2.2? Do you need to time-date stamp your files? Do you need to use password access to files and/or disks? Do you need a speed-up of 2 to 3 times in disk intensive programs? Do you need a little more space in the TPA (say about 4-6k)? Do you need an up-to-date version of SID, MAC, RMAC or LINK? Do you need I/O device redirection, or the ability to add a hard disk drive easily later? How about adding UNIX-like filters or pipes to your CP/M programs? If any or all of these reasons fit your case, or if you love to experiment, then read on. Otherwise save your time.

WHAT HARDWARE YOU NEED

To install a useful version of CP/M 3.0 on your Micropolis-based system, you will need the capability of handling banked memory. This will require a memory management unit or the built-in equivalent such as found on a CompuPro 8085/88 CPU. The memory is divided between global and banked. The global memory (which includes the Micropolis controller) is addressable no matter which bank is switched in. Global memory should occupy the top 16-32k of the 64k of 8080 address space. The banked memory must occupy ALL of the remaining address space in low memory for banks 0 & 1, as a minimum. All other banks may be smaller but must not extend into the global memory.

For example, I am using 32k of global memory. It is a CompuPro RAM 20 (32k @12Mhz) with the base address set at 8000h. The top 4k (F000) is disabled to allow for the OS in ROM (F000-F3FF), the Micropolis controller at F400, and some scratch RAM from F800-FFFF. The banked memory is a RAM-22 (256k @12Mhz) modified to appear as eight banks of 32k each with a base address of 0000h.

Your computer should run at 4-8 Mhz. This is not necessary to run CP/M 3, but the speed advantages of buffered disk access will be hidden by a 2 Mhz clock rate. Be sure the speed jumper is installed on your Micropolis controller (see section 2.1.4.2 Micropolis User's Manual).

WHAT SOFTWARE YOU NEED

Besides the above hardware requirements, you will need a copy of DRI's generic CP/M Plus. This is available, at last report, from Colonial Data Systems, 80 Picket District Road, New Milford, CT 06776, (203) 355-3178. It will come on two 8" SS SD diskettes (IBM format). You must get someone like DAMAN to down-load the two disks onto a pair of Micropolis Mod II diskettes.

CP/M 3.0 (CONTINUED)

You will also need the MUG library diskette containing the programs listed below. These have been written specifically for the Micropolis version of CP/M 3.

DBIOS.ASM - The Micropolis disk driver module.
CBIOS.ASM - A character I/O driver module.
DRVTL.ASM - Micropolis substitute for DRI version.
COPY3.ASM - An utility customized for CP/M 3 on Micropolis.
MDOSDVR.LIB - A library file of Micropolis equates.
LDRBIOS.ASM - Customized part of the system loader.
COLDBOOT.ASM - What Micropolis expects on sector 0, track 0.
COPYSYS.ASM - Debugged version of DRI version (similar to SYSGEN).
MOVE.ASM - Bank to bank moves done by this module.

Also there is a file called MELOS.LIB which holds the I/O entry points and scratch memory addresses. MELOS is a combination of EXTCCP7, SOLOS and some special features I need because I can only type with one hand. If anyone is interested, I will be happy to supply the source code.

You will need a working version of CP/M 2.2 on your system. With CP/M 1.4 you do not have the random disk read/write needed to run GENCPM. GENCPM is a DRI program supplied with CP/M 3 and is used to generate the CPM3.SYS file which is the actual CP/M 3 operating system.

One last thing, you will need some assembly language programming experience. Not a lot, but some. The majority of the work is done but you must customize it to fit your console and memory control hardware.

HOW CP/M 3 IS PUT TOGETHER

The CP/M 3 system program does not reside on the first two disk tracks as did its ancestors. Instead, the Micropolis PROM (at F400h for example) moves a part of its contents to low memory (A0h) where it is executed. This short program reads drive 0, track 0, sector 0. In that sector is found the program COLDBOOT. The COLDBOOT program reads the rest of the first two tracks into memory starting at 400h. This program is called CPMLDR and is made up of a DRI supplied BDOS and a BIOS from LDRBIOS.ASM. The CPMLDR then finds the CPM3.SYS file on drive 0 loads it into high memory in banks 0 and 1. Control is then passed to CPM3.SYS which finds the CCP.COM file on drive 0 and loads it at 100h of bank 1 and makes a copy in bank 0. Control then passes to CCP which puts out the standard A>.

CPM3.SYS contains a resident and a banked portion of the system. It is created by GENCPM from

BNKBDOS3.SPR, RESBDOS3.SPR (both supplied by DRI) and BNKBIOS3.SPR (which you make). The BNKBIOS3.SPR file is the result of linking a set of individual BIOS modules. The modules needed are BIOSKRNL, BOOT, SCB, CBIOS, MOVE, DVRTBL, and DBIOS. Using the RMAC assembler that comes with the CP/M 3 system, the ASM files are assembled into REL files. Then the LINK program is used to tie everything together into a SPR file.

The first step is to make a back-up of the two DRI disks (that is, of their Mod II copies), and of the library disk containing the customized programs. Next make a CP/M 2.2 working disk with SYSGEN, PIP and ED (or your favorite editor) on it. Now copy on to this working disk the following files from the DRI disks:

RESBDOS3.SPR	GENCPM.COM	CPMLDR.REL
BNKBDOS3.SPR	RMAC.COM	LINK.COM
BIOSKRNL.ASM	CCP.COM	MAC.COM
SCB.ASM	BOOT.ASM	CPM3.LIB
MODEBAUD.LIB		

Next copy to your working disk all of the files on the Library disk. Now make a backup of your working disk and start editing the changes described in the following paragraphs as required by your system.

THE BIOSKRNL MODULE (Appendix E)

DRI very thoughtfully wrote a complete set of BIOS modules and put the source files on the CP/M 3 disks and in the "CP/M PLUS SYSTEMS GUIDE" Appendix E through I. These programs provide good examples of the requirements of the BIOS and by using all or part of them we can save an enormous amount of coding and debugging. The first of these is called BIOSKRNL.ASM and appears in Appendix E. As the name implies, it is the core or KERNAL of the CP/M 3 BIOS. It contains the BIOS jump table and much of the code that is not hardware dependent. Note that there are more BIOS entry points than in CP/M 2.2. The only possible change to the module is near the beginning. Be sure that the 'banked equ' statement is set 'true'.

THE SCB MODULE (Appendix F)

The System Control Block (SCB) is a 100-byte-long block of data that is shared by the system and the user. It contains a wide variety of information. The current time and date are kept in the SCB if time/date stamping is implemented. The upper limit of the TPA is saved in the SCB along with the terminal width and height. No changes are needed so use the file just as DRI delivered it.

THE BOOT MODULE (Appendix I.1)

The Boot module does cold and warm booting, sets up page zero, issues the sign-on message and loads and/or reloads CCP. Be sure that the 'banked equ' statement is set 'true'. Change the sign-on message to read:

```
signon$msg: db home,lf,'CP/M Version 3.0'
            db cr,lf,'Ported for Micropolis'
            db cr,lf,'(c) M. M. Dalton, 1985'
```

Note the 'home' does a screen clear and cursor home function. Change the corresponding equate in MELOS.LIB to the code required in your system (for example, a ctrl-k is needed by a Sol-20).

Last but not least, if you have ONLY two banks (0 & 1), then change the code at ld\$1: to

```
ld$1:
...
...
mvi a,0 ! call ?bnksel ; select bank 0
...
etc.
```

and at rl\$1: to

```
rl$1:
mvi a,0 ! call ?bnksel ; select bank 0
...
etc.
```

OTHERWISE leave it as is.

THE CBIOS MODULE (compare Appendix I.2)

This is a simplified version of the CHARIO.ASM file supplied by DRI. It is tailored to use MELOS for communication with the Televideo-950 and the Epson MX-80. Your existing BIOS routines can be transplanted as is in most cases. You may have a system similar enough that changing the equate values in MELOS.LIB will solve the problem. Customizing the CBIOS module is the most difficult part of putting CP/M 3 on your computer.

After studying CBIOS.ASM, CHARIO.ASM, MELOS.LIB, your CP/M 2.2 CBIOS and your computer's OS in PROM (most computers execute some kind of code at power-on time from a built-in PROM), proceed generally as follows:

1. Do your terminals, modems or printer (including their I/O channels) require any initialization that was not done at power-on? Do not include initialization required to do console output (this will have been done by the LDRCPM program). Put any code needed at ?cinit: and end with a ret.

2. Put the code needed to get a character from the keyboard at ci0:. It must be in the 'A' register with the parity bit striped and the flags set.

3. The code to check keyboard status goes after the jnz cist1 in the ?cist: routine. It must not destroy the data byte. The 'A' register must hold FFh if a byte is ready, otherwise a zero is returned with flags set.

4. The code to send a byte from 'C' to the console goes after the jnz col: in the ?co: routine.

5. The code to send a byte from 'C' to printer goes at label centronics\$out: and ends with a ret.

6. If either your console or printer can be tested for output ready status, the code goes in place of 'ori FFh ! ret' in ?cost: or cent\$stat: respectively. If not, leave the code as is.

7. The data structure @ctbl: must remain but the device names can be changed (each name must be six characters long, i.e., pad on the right with blanks as needed). The baud\$19200 or baud\$300 can be changed to whatever your actual baud rates are. This data is used by a CP/M 3 utility called DEVICE and does not directly effect CP/M 3 itself.

THE MOVE MODULE (compare Appendix I.5)

This module is a replacement for the DRI module of the same name. It provides for moves within a bank, between banks, and bank selection. The only change you may need is in the ?bank: routine. On entry to this routine the 'A' register holds the new bank number in binary (i.e., 0=00h, 1=01h, 2=02h, 3=03h, etc.). The routine switches this new bank into operation without changing the contents of any register but 'A'. In the case of my CompuPro 85/88 CPU, all I have to do is an 'out bank'. MELOS.LIB contains an equate setting 'bank' to FDh. You may need to change this equate value to whatever your bank select port is set to. If your memory management hardware requires a 1 of 8 code in 'A' (i.e. 0=00h, 1=01h, 2=02h, 3=04h, 4=08h, etc.), then you will have to devise some code to insert at ?bank: which saves any other registers you need, changes the value from binary to 1 of 8 and restores the registers to their original values.

THE DVRTBL MODULE (compare Appendix I.3)

This module replaces the DRI module of the same name. It is a list of the disk driver entry points in DBIOS of each drive in your system. I have set it for a four drive system. If you have less, do not worry.

Should you accidentally ask for a non-existent drive, CP/M 3 will return an error message to the console "BIOS error on D: T=00002, S=00000, Read, Drive not up, Retry (Y/N) ?". Respond with a 'N' and you will be safely back at the CP/M 3 command level.

THE DBIOS MODULE (compare Appendix I.4)

This module has all of the data structures and code required by CP/M 3 and Micropolis to do disk read or

write operations. It tells the BDOS all about the number of tracks, sectors per track, bytes per sector, and sector skew. Transfers are made one sector at a time. The BDOS handles all deblocking. Although there are 268 bytes per Micropolis sector, CP/M (all versions) use only the last 256 bytes. Therefore, it is convenient to double buffer the data to and from the disk. During a disk read, for example, 268 bytes are read into a special buffer and then the last 256 bytes are sent to the DMA address specified by the BDOS. The address of this buffer is specified in MELOS.LIB. It should be as high in the global memory space as possible. In my case, I have 2k of scratch memory starting at F800h (just above the Micropolis controller). Thus the MELOS.LIB entry is 'mdosbuf equ F880h'. If this area in memory does not exist in your system or is otherwise occupied, then find a new home for 'mdosbuf'. If nothing else is available, then reserve the top two pages of RAM in global memory and change the MELOS.LIB equate to put 'mdosbuf' at the beginning of the two reserved pages.

Two other pieces of data used by DBIOS, but shared by other programs, must have safe homes which must be predefined in MELOS.LIB. They are 'xdadr' and 'xmode'. If you are reserving two pages for 'mdosbuf' then there is room for these two items at the end of the second page. Two bytes are required by 'xdadr' (the address of the Micropolis controller) and one byte is needed by 'xmode' (used by DBIOS to inhibit the pre-write header check and/or the postwrite verify). COLDBOOT initializes 'xdadr' and 'xmode'. COPY3 sets and resets 'xmode' as needed to format a fresh disk.

THE LDRBIOS MODULE

The LDRBIOS.ASM file is a stripped version of the CPM3 BIOS. It provides for console output, disk read and any initialization required. The only part that requires attention is the console output routines. They should be changed to match the ones you are going to use in CBIOS.

THE COLDBOOT MODULE

The COLDBOOT.ASM file is the source of the code needed in sector 0 of track 0 of the default drive (i.e. drive 0 or A). Only one change is needed on any system using a Micropolis controller. There is an error exit address called 'XCOLD' which is defined in MELOS.LIB. COLDBOOT will take this exit if there is faulty or no memory for the CPMLDR program as it is being loaded from tracks 0 & 1. 'XCOLD' in MELOS must equal the address your computer goes to on a manual reset or power-on-jump.

PUTTING IT TOGETHER

Back-up your working disk again and review all the changes you have made before proceeding. Errors found now are errors that won't plague you later.

When you are satisfied, start assembling the individual files.

Assuming that all the files discussed so far are on a disk in drive A, then execute the command line

```
RMAC <filespec> $AA SZ RA LA PZ
```

on the filespecs BIOSKRNL, SCB, BOOT, CBIOS, DVRTBL, MOVE, DBIOS & LDRBIOS.

Any errors the assemblers report must be corrected before going on. You should now have a REL file for each assembled file.

The next step is to link the modules together. Execute the following link commands:

```
LINK BNKBIOS3[B]=BIOSKRNL,SCB,BOOT,  
CBIOS,MOVE,DVRTBL,DBIOS
```

and

```
LINK CPMLDR[L100]=CPMLDR,LDRBIOS
```

You will now have a BNKBIOS3.SPR file and a CPMLDR.COM file. Now you are ready to execute the GENCPM command. It is self prompting and is described in detail in Section 5 of the CP/M 3 SYSTEM GUIDE. The answer to each question is either Y, N or a HEXEDECIMAL number. If you wish to use a decimal number, be sure to prefix the number with a # sign (#22 is decimal while 22 is hexadecimal).

The question "top page of memory (FF) ?", requires a bit of thought. It is asking you to tell it the highest numbered page in global RAM that CP/M 3 can use. Global memory must be contiguous. In other words it can not be interrupted by ROM (like the Micropolis controller) or missing memory. Thus the top page cannot be above your Micropolis controller address (F400 ?). If your DBIOS module uses 2 pages at the top of your supply of RAM for scratch, then your answer must refer to the page just below them.

Also, remember which bank your boot program is going to use to hold the copy of CCP. It will be in bank 0 if you only have two banks. If you have more banks, then it will be in bank 2. When GENCPM asks "Enter memory segment tables", you must specify (for the appropriate bank) a base of 0D and reduce the size by 0D hex pages. This will allow room at the bottom of the bank for a copy of CCP.

When GENCPM signs off, you will have a CPM3.SYS file on your working disk. Now is a good time to back-up the disk again. After that precaution, you are ready for the big moment. Execute CPMLDR. You SHOULD see the CP/M V3.0 Loader sign-on message followed by the memory load map and TPA size. This will be followed by the System sign-on and the friendly A>. You are now running CP/M 3!!!!!! Try it out to be sure it talks to the console, printer and can load or save data on a disk. Remember that your CP/M 2.2

system has been over-written in your computer memory. Be sure to coldboot it back into memory if there are any problems.

If the loader fails to sign-on, the problem is in the console output routine in LDRBIOS.ASM. Fix it and be sure to make the same corrections to CBIOS.ASM. Then re-assemble those two files and re-link everything. Repeat GENCPM and try again. The next most likely problem areas are the console input and status routines or the bank select code. To find these problems you will need to create an unbanked version.

Change the 'banked equ true' statements in BIOSKRNL and BOOT to 'banked equ false'. Reassemble these two modules and then execute:

```
LINK BIOS3[OS]=BIOSKRNL,SCB,BOOT,
CBIOS,MOVE,DVRTBL,DBIOS
```

Execute GENCPM and specify a non-banked version. Also, refer to Section 6 of the CP/M 3 System Guide on how to determine the size to use and how to use CP/M 2.2 and SID or DDT to debug the remaining problems.

THE LAST STEP

Now that everything is working, it is time to put COLDBOOT and CPMLDR on the 'system' tracks of a disk. This will allow you to coldboot directly into CP/M 3. Execute the following to create the files needed:

```
LINK CPMLDR[L400]=CPMLDR,LDRBIOS
and
MAC COLDBOOT $AA $Z $HA $LA $PZ
```

Next execute SID (or DDT) without a command tail and proceed with the following dialog:

```
#F100,3000,0
#ICPMLDR.COM
#R900
#ICOLDBOOT.HEX
#R
#M16B,26A,900
#FB25,B3D,0
#SB3D
0B3D 00 <x> <x> = CTRL code which clears
                        your CRT screen.
0B3E 43 .
#GO
```

Execute SYSGEN and skip the source read (you already have it in memory). Choose a drive that doesn't have your working copy in it, insert a freshly formatted disk and specify that drive as the destination. The new disk now has COLDBOOT and CPMLDR on tracks 0 & 1. Copy CPM3.SYS and CCP.COM to the new disk. Place the new disk in drive A, reset your system and cold boot it. CP/M 3 should sign-on with no difficulty.

LOOSE ENDS

ERASE the CPMLDR.COM file. The version that was created by the linker with a [L400] option will not execute but was needed by COLDBOOT on the system tracks. Use MAC to assemble COPYSYS.ASM and COPY3.ASM into COM files. COPYSYS.COM is the CP/M 3.0 version of SYSGEN. It does the same basic job with a few new bells and whistles. COPY3.COM allows for formatting a raw disk or it will copy all tracks from one disk to another previously formatted one. COPY3 requires knowledge from MELOS.LIB of where you told COLDBOOT to put 'xdadr' and 'xmode'.

Last but not least, use CP/M 2 PIP to copy GENCPM.COM, INITDIR.COM, HELP.HLP and the file which contains a version of this article, CPM3.DOC, to a CP/M 3 disk and then use DDE or a similar disk editor to change the directory entries so that they conform to CP/M 3.0 standards. See DDE.DOC for the changes needed and how to use DDE (DDE is an MDOS program).

Editor's note: The amount of work that Mel put into this article is hard to comprehend. I surely appreciate his sharing the experience with us.

The library programs mentioned are on MUG CP/M disk 1918, except for DDE, which is on MUG MDOS disk 908.

MORE ON THE SURVEY (cont. from pg. 1)

A couple other useful statistics can be calculated from the data.

VG-all models	41.0%
Micropolis-based	41.0%

The above statistics are by computer. Another way to look at things is by member. As one would expect, 97.4% of the members have a Micropolis or Vector Graphic computer. Members who have only one computer equal 43.6%, while 56.4% have more than one computer. The above calculated average of 2 computers per member is just that, only an average. Obviously, some members have *considerably more* than one computer.

A somewhat interesting statistic is that 79.5% of the members have at least one computer that is Micropolis based. A Micropolis-based computer is one that either has external Micropolis drives, or is a Vector MZ or Vector System-B.

Of the people who have more than one computer, 63.6% have an IBM-type machine. For the total membership, 35.9% have an IBM-type machine.

Interpreting statistics is always a risky business, but I'll hazard the guess that this means most members are

MICROPOLIS/VG USERS GROUP

Newsletter #73
August 1986

tending to use their Micropolis/Vector machines less, and their IBM machines more.

Finally, only 5.1% of the members have a lap computer (the TRS-100s). I expected a larger percentage.

NEXT MONTH

With a little luck, it won't take 13 months to discuss all the questions in the survey. But then ... Some pretty interesting stuff here. Stay tuned. We'll try and see what the MUG can do to respond to the requests of 'what a user group should do'.

MUG BULLETIN BOARD?

Matthew Lewis has expressed his desire to maintain a bulletin board for the MUG. He has all the hardware and is looking for one or more other members to write the software. He would prefer the program be written in MDOS Basic. This would allow him to get it running under OSM on his 8" winchester hard disk.

Some of the benefits of this board would be access to the library catalogs. We could even put the members & their H/W & S/W on (with their approval). Then, if you had a particular problem, you could find out who has the same H/W or S/W and contact them for help.

If you are interested in helping Matthew, please contact him at 4327 Lloyd, Kansas City KS 66103, phone (913) 831-1755.

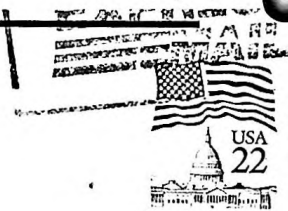
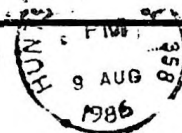
CLASSIFIED

The Classified Section of the MUG Newsletter is available to all users, free of charge. Just call or write us with a description of what hardware, software, information, etc., that you want to obtain, or dispose of.

WANTED - Vector Graphic B system in working condition.

Contact Elliott Automation, Ken or Peter, at (713) 530-2108 days, or Peter at (713) 465-3797 nights.

MICROPOLIS/VECTOR GRAPHIC
USERS GROUP
A Division of **DAMAN**
604 Springwood Circle
Huntsville AL 35803-1740
(205) 881-1697



FIRST CLASS MAIL

FIRST CLASS MAIL